AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated in the following listing of all claims:

1. **(Currently Amended)** In a microprocessor that manages interlock conditions for load instructions and that supports speculative and out-of-order execution of load instructions, a method of bypassing data to younger instructions, comprising:

identifying a source register upon whose contents a current instruction depends for execution;

providing a load annex, the load annex having a plurality of sequential load entries, each of the plurality of sequential load entries being capable of storing data associated with the source register, wherein each load entry is associated with one of a plurality of sequential priority levels, the plurality of sequential priority levels comprising a highest priority level and a lowest priority level;

storing, in the sequential entries of the load annex, a plurality of load data associated with a particular register, wherein the plurality of load data associated with a the particular register are stored in program order with respect to each other;

providing a non-load annex, the non-load annex having a plurality of non-load entries, each of the plurality of non-load entries being capable of storing data associated with the source register, wherein each entry of the non-load annex is associated with one of the plurality of sequential priority levels;

providing that, at most, only one among the load entry associated with a particular priority level and the non-load entry associated with the particular priority level contains a value associated with the source register;

locating, among the plurality of load annex entries and the plurality of non-load annex entries, a freshest value associated with the source register; and

providing the freshest value for use in execution of the current instruction.

2. (Original) The method recited in Claim 1, wherein locating a freshest value further includes:

determining whether any annex entry associated with the highest priority level contains a value associated with the source register, and, if so, selecting the value in the annex entry associated with the highest priority level as the freshest value; and

- 4 -

determining, if no annex entry is associated with the highest priority level, whether any annex entry associated with the lowest priority level contains a value associated with the source register, and, if so, selecting the value in the annex entry associated with the lowest priority level as the freshest value.

3. (Original) The method recited in Claim 1, wherein:

the plurality of priority levels includes at least one intervening level between the highest priority level and the lowest priority level.

4. (Original) The method recited in Claim 3, wherein locating a freshest value further includes:

determining whether an annex entry associated with the highest priority level contains a value associated with the source register, and, if so, selecting the value in the entry associated with the highest priority level as the freshest value;

if not, sequentially determining whether an annex entry associated with successively lower priority levels lower than the highest priority level contains a value associated with the source register, selecting as the freshest value the value associated with the source register from the annex entry with the highest of the lower priority levels.

5. (Original) The method recited in Claim 1, further comprising:

providing a plurality of non-load annexes.

6. **(Currently Amended)** A computer system that bypasses load data to younger instructions, comprising:

a main memory;

at least one processing unit coupled to the main memory, the processing unit, being configured to execute load instructions and also being configured to execute a current instruction, having a source register;

a load annex, the load annex having a plurality of sequential load entries, each of the plurality of sequential load entries being capable of storing data associated with the source register, wherein each load entry is associated with one of a plurality of

- 5 -

sequential priority levels, the plurality of sequential priority levels comprising a highest priority level and a lowest priority level;

a scoreboard that manages interlock conditions for the load instructions executed by the processing unit, wherein the scoreboard further comprises a module that permits execution of a plurality of load instructions having the same destination register, wherein the plurality of load instructions having the same destination register are executed in program order with respect to each other;

a non-load annex, the non-load annex having a plurality of non-load entries, each of the plurality of non-load entries being capable of storing data associated with the source register, wherein each entry of the non-load annex is associated with one of the plurality of sequential priority levels;

a module, coupled to the main memory, that provides for locating, among the plurality of load annex entries and the plurality of non-load annex entries, a freshest value associated with the source register; and

a module that is operable to provide the freshest value for use in execution of the current instruction.

7. **(Currently Amended)** The computer system recited in Claim 6, wherein:

the module ~~the~~ <u>that</u> provides for locating a freshest value further includes:

a module that is operable to determine whether any annex entry associated with the highest priority level contains a value associated with the source register, and, if so, is further operable to select the value in the annex entry associated with the highest priority level as the freshest value; and

a module that is operable to determine, if no annex entry is associated with the highest priority level, whether any annex entry associated with the lowest priority level contains a value associated with the source register, and, if

so, is further operable to select the value in the annex entry associated with the lowest priority level as the freshest value.

8. (Original) The computer system recited in Claim 6, wherein:

- 6 -

the plurality of priority levels includes at least one intervening level between the highest priority level and the lowest priority level.

9. **(Currently Amended)** The computer system recited in Claim 8, wherein:

the module ~~the~~ <u>that</u> provides for locating a freshest value further includes:

a module that is operable to determine whether an annex entry associated with the highest priority level contains a value associated with the source register, and, if so, is further operable to select the value in the annex entry associated with the highest priority level as the freshest value; and

a module that is operable, if no annex entry associated with the highest priority level contains a value associated with the source register, to sequentially determine whether an annex entry associated with successively lower priority levels contains a value associated with the source register, and, if an entry at one or more priority levels lower than the highest priority level contains a value associated with the source register, is operable to select as the freshest value the value associated with the source register from the annex entry with the highest of the lower priority levels.

10. **(Original)** The computer system recited in Claim 6, further comprising:

a plurality of non-load annexes, each of the plurality of non-load annexes having a plurality of non-load entries, each of the plurality of non-load entries being capable of storing data associated with the source register, wherein each entry of each non-load annex is associated with one of the plurality of sequential priority levels.

11. **(Currently Amended)** A computer system, comprising:

a main memory;

at least one processing unit coupled to the main memory, the processing unit, being configured to execute load instructions and also being configured to execute a current instruction, having a source register;

means for identifying ~~a~~ <u>the particular</u> source register upon whose contents a current instruction depends

- 7 -

for execution;

means for storing a plurality of load data associated with the source register wherein each load data is associated with one of a plurality of sequential priority levels, the plurality of sequential priority levels comprising a highest priority level and a lowest priority level;

means for storing a plurality of load data associated with a particular register in program order with respect to each other;

means for storing a plurality of non-load data associated with the source register, wherein each non-load data is associated with one of the plurality of sequential priority levels;

means for providing that, at most, only one among the load data associated with a particular priority level and the non-load data associated with the particular priority level contains a value associated with the source register;

means for locating, among the plurality of load data and the plurality of non-load data, a freshest value associated with the source register; and

means for providing the freshest value for use in execution of the current instruction.

12. (Original) The computer system recited in Claim 11, wherein means for locating a freshest value further comprises:

means for identifying whether any load data or non-load data associated with the highest priority level contains a value associated with the source register, and, if so, selecting the identified data as the freshest value; and

means for identifying, if no annex entry is associated with the highest priority level, whether any load data or non-load data associated with the lowest priority level contains a value associated with the source register, and, if so, selecting the identified data as the freshest value.

13. (Original) The computer system recited in Claim 11, wherein:

the plurality of priority levels includes at least one intervening level between the highest priority level and the lowest priority level.

14. (Original) The computer system recited in Claim 13, wherein means for locating a freshest value further includes:

> means for identifying whether a load data or non-load data associated with the highest priority level contains a value associated with the source register, and, if so, selecting the identified data as the freshest value;

> means for, if no load data or non-load data associated with the highest priority level contains a value associated with the source register, sequentially identifying whether a load data or non-load data associated with successively lower priority levels contains a value associated with the source register, and, if an entry at one or more priority levels lower than the highest priority level contains data associated with the source register, selecting as the freshest value the identified data associated with the highest among the lower priority levels.

15. (Original) The computer system recited in Claim 11, further comprising:

> providing a plurality of means for storing a plurality of non-load data associated with the source register, wherein each non-load data is associated with one of the plurality of sequential priority levels.